

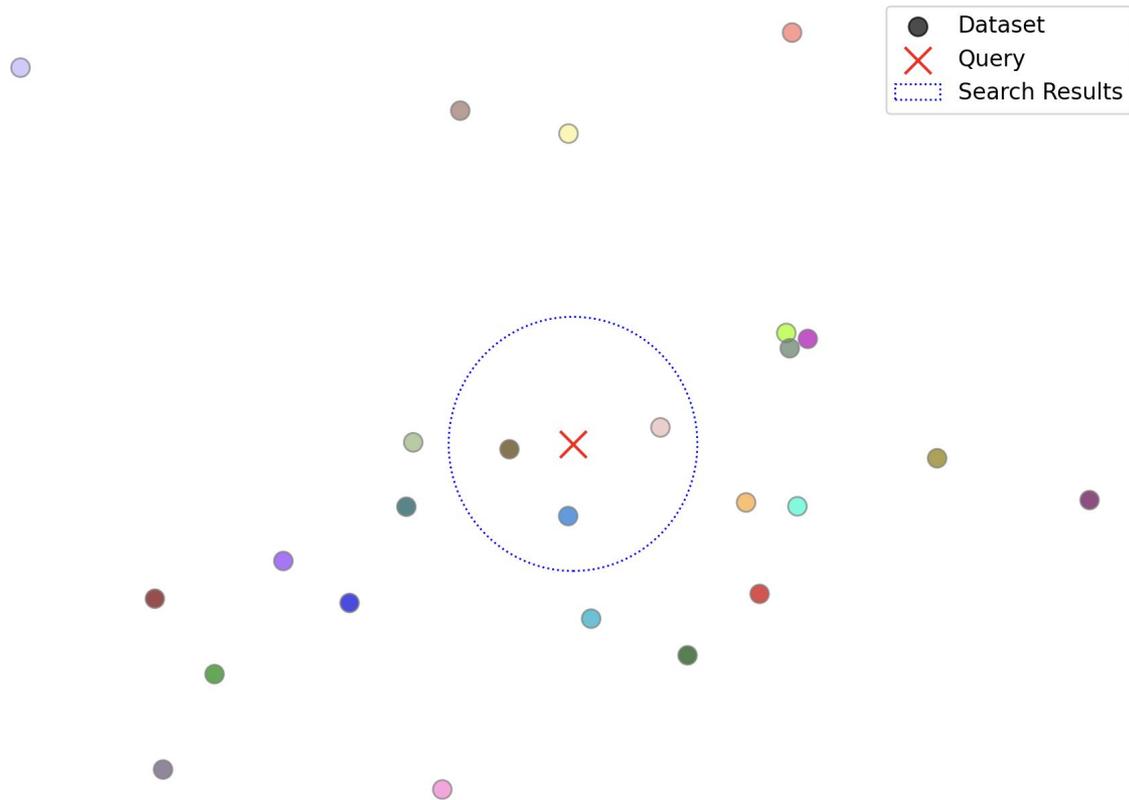
DESSERT: An Efficient Algorithm for Vector Set Search with Vector Set Queries

By Joshua Engels, Benjamin Coleman, Vihan Lakshman,
Anshumali Shrivastava



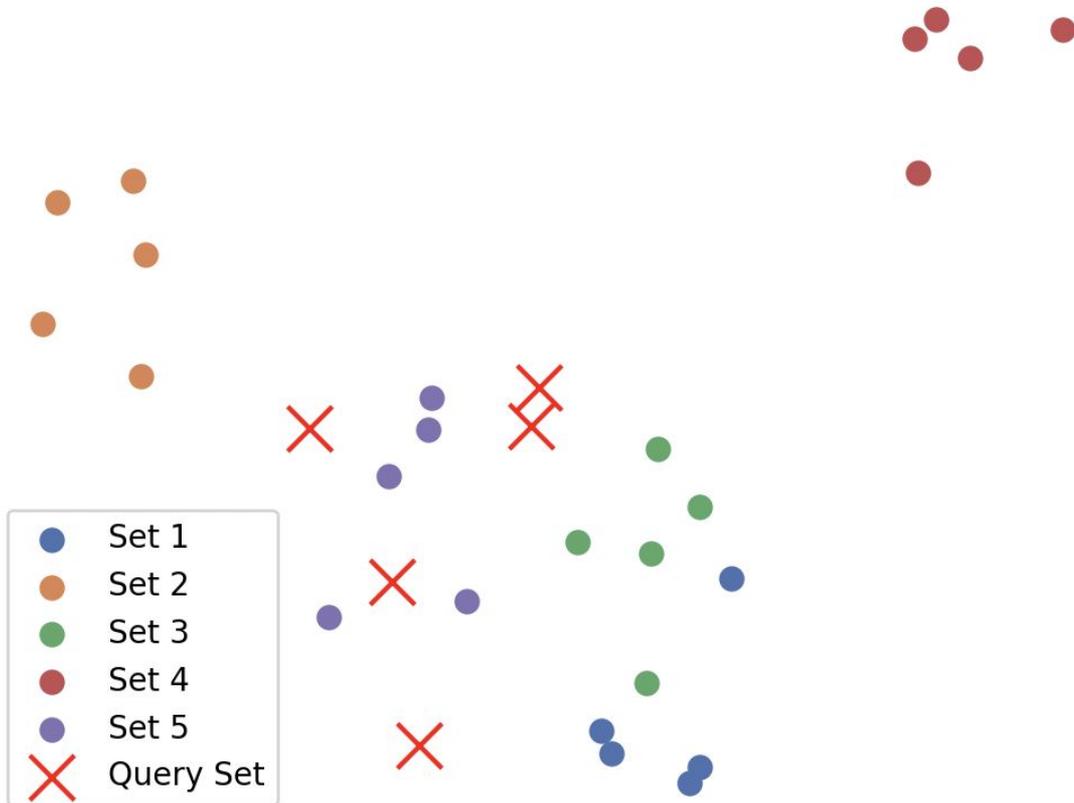
ThirdAI

Traditional Vector Search



Vector Set Search

Less obvious which set is the most similar to the query



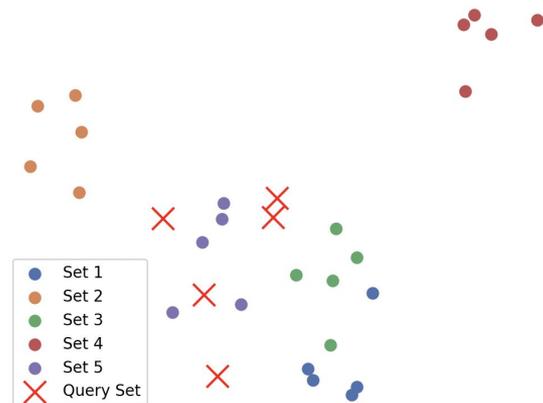
Vector Set Search Problem

- Given set-to-set similarity function $F(X, Y)$
- E.g.

$$F(X, Y) = \sum_{x \in X} \left[\max_{y \in Y} (\text{sim}(x, y)) \right]$$

- Given N vector sets S_i and query set Q
- **Find:**

$$S^* = \underset{i \in \{1, \dots, N\}}{\operatorname{argmax}} F(Q, S_i)$$

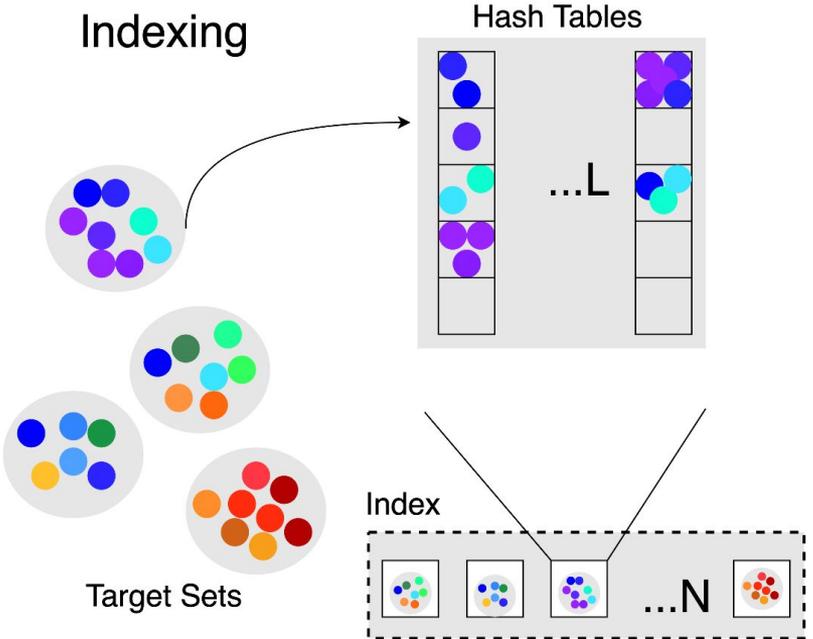


Why do we care?

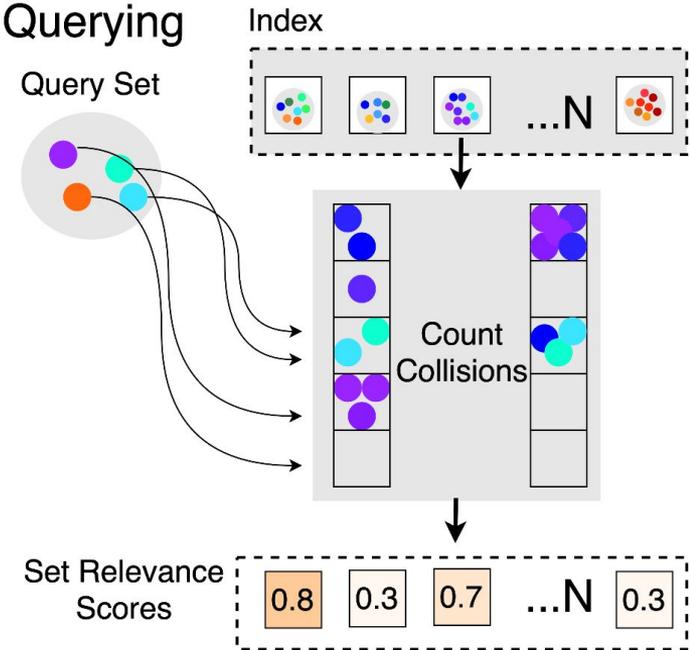
- Many objects can be better represented as collections of vectors!
- Document search: embed each word in each document
 - CoBERT (PLAID) gets SOTA
- Problem: No existing efficient approximate algorithm like for single vector search

The image shows a Google search interface with the query "how do information retrieval systems work". The search results are displayed in a list format. The first result is from upGrad, titled "Information Retrieval System Explained: Types, Comparison ...". The second result is from GeeksforGeeks, titled "What is Information Retrieval?". The third result is from Wikipedia, titled "Information retrieval". The fourth result is from Quora, titled "What are the components of the information retrieval system?". The fifth result is from Coveo, titled "What is Information Retrieval?". Each result includes a snippet of text and a link to the source.

Solution: Dessert



Querying



Theoretical Query Latency

- Bruteforce: $O(m_q m N d)$

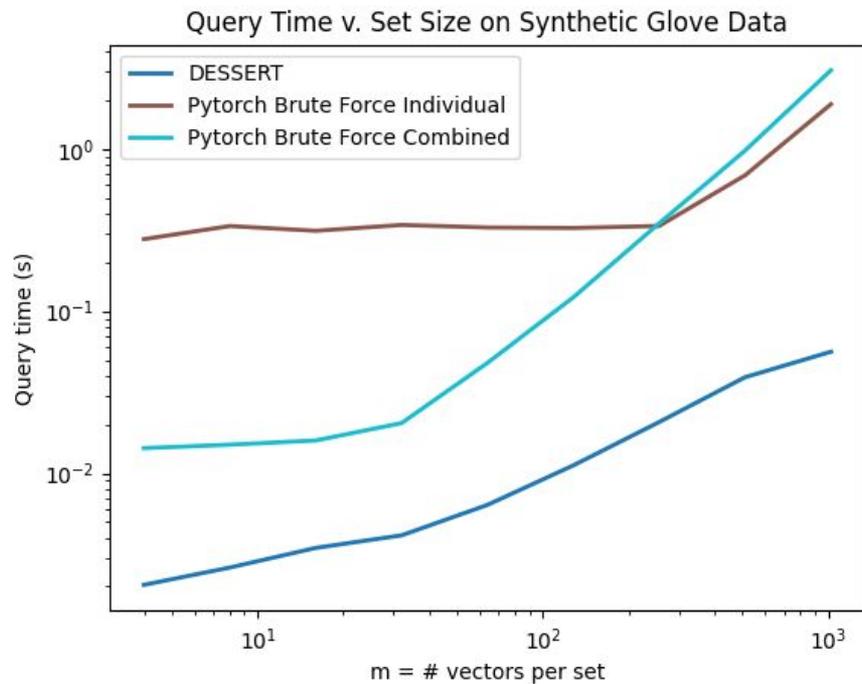
- Ours: $O(m_q \log(N m_q m / \delta) d) + m_q N \log(N m_q m / \delta)$

- $1 - \delta$ is probability of success
- Elided dataset dependent constants
- $1 - \delta$ inversed dependence on m and d

One time hashing cost,
usually negligible

Cost to query all N sketches with
each of the m_q query vectors

Empirical Results: Synthetic Data



Empirical Results: MS MARCO (passage retrieval)

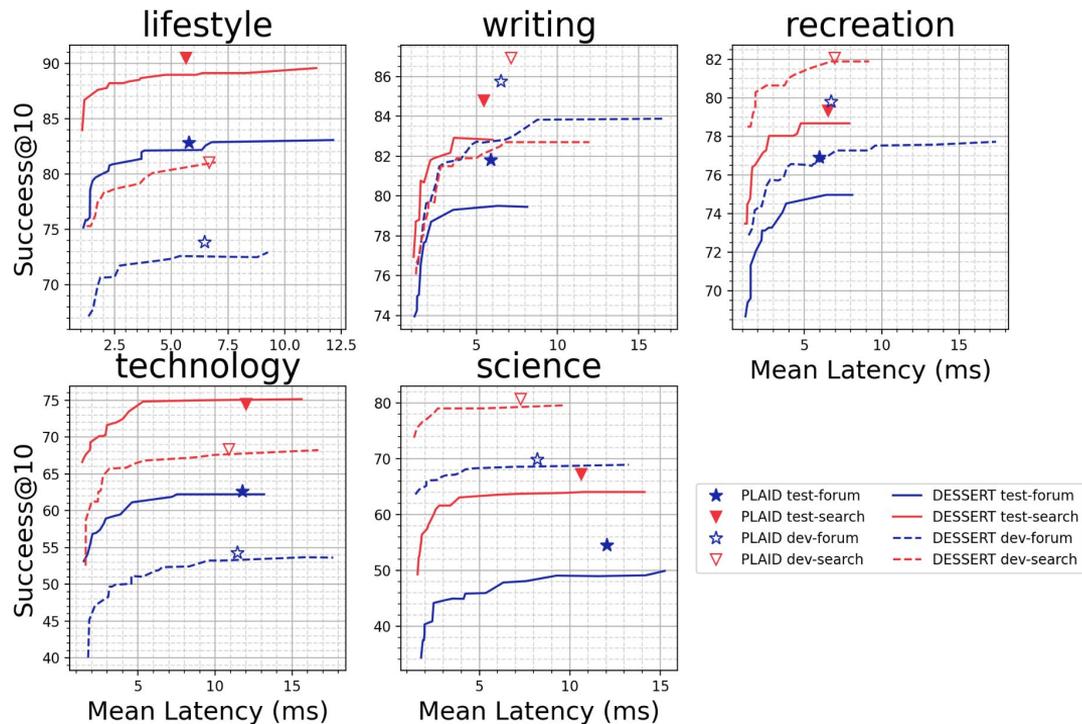
Results optimized for k = 10:

Method	Latency (ms)	$MRR@10$
DESSERT	9.5	35.7 ± 1.14
DESSERT	15.5	37.2 ± 1.14
PLAID	45.1	39.2 ± 1.15

Results optimized for k = 1000:

Method	Latency (ms)	$R@1000$
DESSERT	22.7	95.1 ± 0.49
DESSERT	32.3	96.0 ± 0.45
PLAID	100	97.5 ± 0.36

Empirical Results: LoTTE (passage retrieval)



Future Work

- Can we build a vector-set search algorithm that is sublinear in N ?
- What other data domains beyond passage retrieval can we try?
 - Image search: use pre-pooled embeddings
 - E-commerce: embed each item in a basket
 - Clustering: find most similar cluster to existing cluster
- What additional problems can be sped up with approximate set similarity calculations?

Team and Contact

Josh Engels: josh.adam.engels@gmail.com

Previously at ThirdAI, now a PhD student at MIT

Ben Coleman: benjamin.ray.coleman@gmail.com

Previously at ThirdAI, now a research scientist at Google DeepMind

Vihan Lakshman: vihan@thirdai.com

AI Engineer at ThirdAI

Anshumali Shrivastava: anshumali@rice.edu

CEO of ThirdAI, associate professor at Rice University

